Deriving grid workload models from user submission strategies

Diane Lingrand[1], Johan Montagnat[1] and Tristan Glatard[2]
[1] MODALIS team - I3S Univ. Nice - Sophia Antipolis / CNRS - FRANCE
{lingrand,johan}@i3s.unice.fr
[2] University of Lyon, Creatis-LRMN - FRANCE
glatard@creatis.insa-lyon.fr

November 30, 2009

Abstract

Production-grid users experience many system faults as well as high and variable latencies due to the scale, complexity and sharing of such infrastructures. To improve performance, they adopt different submission strategies, that are potentially aggressive for the infrastructure.

This work studies the impact of three different strategies. It is based on a probabilistic modeling of these strategies which are evaluated according to their performance, measured as the reduction of the latency expectation, and the infrastructure overhead, measured as the additional number of submitted jobs. A strategy cost criterion is then derived.

Experiments are performed using real workload traces collected from the EGEE production infrastructure. Under these conditions, a good balance between high performance and low overhead can be found.

This research report in an extension of the paper [8] and includes the demonstration of the studied models.

1 Introduction

Production grids are systems world-wide distributed, covering various network domains connected by WANs. They are highly variable environments federating sites with heterogeneous resources, configuration rules and access policies and servicing many users concurrently. Consequently, grids are characterized by high and non-stationary workloads. From a user point of view, the grid appears as a plethora of queues operated through different batch management systems, with different prioritization policies, that are difficult to exploit efficiently.

Unlike supercomputers no single grid scheduler can obtain a global view of the complete grid infrastructure. Instead, meta-schedulers coordinate the action of site schedulers, using heuristic that try to cope with (i) partial information and (ii) local independent scheduling policies that may interfere with the meta-scheduling objective. Moreover, job life-cycles include much more steps than just scheduling and queuing (among others: credentials delegation, match-making, job submission language formats translation, logical-to-physical file names resolution, file replica selection, job monitoring system, grid information system). In the order of 10 machines, that all have to be reachable, may be involved in the job submission process.

The job submission system is subject to failures that can originate from network or connectivity problems, local configuration issues (authorization issues, differences in middleware versions, local environment, etc), workload variations (impacting all the components, not only the job queuing time), data access (data catalog querying, data transfer queuing, and the transfer itself), and scheduling issues. Nobody currently has a proven classification of all possible issues. Such a list is far from being trivial given the heterogeneity of the grid and the interoperation of complex stack of services.

As a consequence, grid jobs *latency*, measured as the duration between the beginning of a job submission and the time it starts executing, can be very high and prone to large variations. High fault ratios are often reported, as for example in [2]. High latency and faults impact the performance of applications, sometimes making grids unworthily for their tasks. Assuming that a single entity could achieve flawless scheduling on a world-wide scale production grid seems quite hazardous and we claim that a client-side error control is required as part of the scheduling, similarly to transport protocols implementing fault-tolerance mechanisms to deal with routing errors on the Internet.

As a matter of fact, it has been commonly observed on production systems that despite the middleware best efforts to address these problems internally, users do adopt different submission strategies controlled on the client side such as canceling jobs with too long latencies before resubmission or submitting several copies of the same job. These strategies are often empirical and not objectively evaluated. The motivation for this work is to provide an objective insight on these strategies. Strategies with a proven impact on application performance and causing no critical overload of the job management system could then be integrated in the client side of the middleware to release the users of this burden.

The more variable the job latency, the more efficient multiple submission strategies. Although multiple submission strategies are effective from the user point of view, they are usually not appreciated by administrators since they are increasing the workload endured by the grid resource brokers and schedulers. Yet, little can be done to regulate their usage, especially on a wide scale distributed grid. Quantifying the impact of submission strategies on real-size grids is a difficult problem due to the complexity of such systems. In this paper, we adopt a probabilistic approach to model grid responsiveness. Execution statistics are collected on the real-scale EGEE production grid over long periods of time to estimate probability laws of the job latencies. Several submission strategies are studied using these grid workload models: the *single resubmission* (section 4), the *multiple submission* (section 5) and a novel strategy that we call *delayed resubmission* (section 6).

These strategies are assessed by taking into account two criteria: the performance improvement from user point of view (reduction in latency time) and the infrastructure load (number of tasks to manage). The delayed resubmission strategy demonstrates a very good balance between high performance and low load.

2 Related work

Many works study the scheduling of jobs on distributed systems at different levels. The algorithms are directly implemented in schedulers. For example, Subramani *et al* [17] compare two different scheduling schemes with respect to the slowdown computed as the ratio of (latency + runtime) over runtime. The first one is the "K-distributed scheme" which consists in submitting each job to the K least loaded sites. When the first job starts running, the other (K-1) jobs are canceled. The second one is the "K-Dual Model" which is an improvement of the K-distributed one. It gives priority to locally submitted jobs, using two different queues. The performance analysis shows that the slowdown is better reduced by the K-Distributed than the K-Dual Queue. However, when considering lightly loaded sites and heavily loaded sites separately, there is an inversion with the K-Distributed scheme which does not occur with the K-Dual Queue one. Different values of K have been studied from 1 to 4, showing a decrease in the average slowdown. The authors also demonstrate that the impact of inaccuracies in user estimates of runtime is in favor of the proposed schemes and that they also perform better than the other schemes when considering communication overheads.

Sabin *et al* [15] study the scheduling in a heterogeneous multi-sites environment. They evaluate different strategies including multiple submissions (k=4), conservative versus aggressive backfilling and the relative job efficacy for queuing priority. Different experimentats based on job traces showed that in the case of heterogeneous multi-sites (which is the case in a production grid), conservative backfilling is better than aggressive and that using the relative job efficacy for queuing priority improves performances.

Casanova [3] studies the multi-submission of jobs, considering that when the first job starts running, all other jobs waiting in queue are canceled. He observes that submitting all jobs several times increases their average performance; the users that are penalized are those that do not use multiple submission. The load on batch schedulers or network will not be critical if the number of multiple jobs is less than 30, assuming that submissions are uniformly distributed among sites and that the job inter-arrival is always 5 seconds. However, the author observes that the 2006 version of GRAM leads to a bottleneck when using more than 3 multiple submissions during peak job submission.

In this paper we focus on large scale infrastructure characterized by highly variable latencies, such as the EGEE production grid[1] on which experiments were conducted [7, 12]. Due to its unique scale, it enables challengingly large applications but it is known

[1] EGEE: http://www.eu-egee.org

to be prone to faults and variable queuing times [10, 2, 1]. Yet, variability has also been acknowledged as a critical factor on other types of platforms, such as the knowARC grid[2]: for instance, the experiment reported in [14] shows variations of up to 40 minutes in the grid time of an embarrassingly parallel application and the work presented in [13] shows overheads ranging from 0 to 45 min. Earlier works conducted before the emergence of large-scale grids in the 2000s already pointed out the importance of variability, such as in [16]: authors were then dealing with seconds-to-minutes variability values and the importance of this issue has dramatically increased in current grid infrastructures.

In order to address the complexity of modeling an heterogeneous grid infrastructure, we adopt a probabilistic approach. Statistical studies carried out in [4] and [6] and earlier works such as [11] or [5] are important contributions to workload modeling. However, those works mostly consider latencies from the perspective of a specific grid scheduler (if not a local site queue) and the actual match with the latency observed by an application still has to be validated. As explained above, several additional steps such as data transfers and proxy delegations are likely to disturb measures carried-out at the grid or local batch scheduler levels. Consequently, the monitoring approach adopted in this work relies on round-trip times of probes submitted to the grid from the user client, thus getting much closer to real application job submission conditions.

3 Definitions, assumptions and reference data

We define the *latency* as the duration between the instant job submission instant and the beginning of its execution on a computation resource. It is modeled through a random variable R. On the EGEE production grid, the latency distribution has been observed to be heavy-tailed, as reported, e.g., in [9]. We denote the *fault* (or *outlier*) *ratio* that commonly occur on a production grid as the real value $\rho \in [0, 1]$. Figure 1 plots F_R, the cumulative density function (cdf) of the heavy-tailed random variable R and $\tilde{F_R}(t) = (1-\rho)F_R(t)$, the cumulative histogram of all latencies (normalized with respect to all submitted jobs, including outliers).

The probability for the latency of a job to be lower than a given threshold t depends on the probability of the job to not be an outlier (probability $1-\rho$) and $F_R(t)$:

$$P(R < t) = (1-\rho)F_R(t) = \tilde{F_R}(t)$$

Conversely, the latency is longer than t if the job is an outlier (probability ρ) or it is not an outlier (probability $1-\rho$) and it terminates before t (probability $1-F_R(t)$):

$$P(R > t) = \rho + (1-\rho)(1-F_R(t)) = 1-(1-\rho)F_R(t) = 1-\tilde{F_R}(t)$$

It has to be noted that although $P(R < t) = \tilde{F_R}(t)$, $\tilde{F_R}$ is not a cdf (it does not converge towards 1) and it cannot be considered as such in the subsequent calculations.

[2] knowARC: http://www.knowarc.eu/

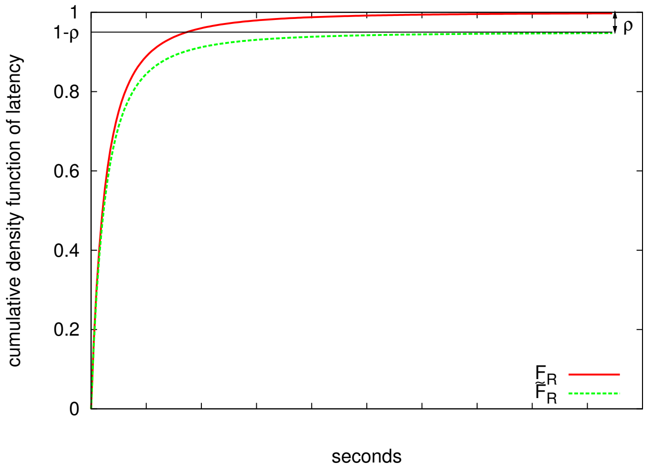

Figure 1: Cumulative density of latency (s)

3.1 Experimental measurements

Our reference experimental data has been obtained on the biomed Virtual Organization (VO) of the EGEE production grid. With 80,000 CPU cores dispatched world-wide in more than 240 computing centers, EGEE represents an interesting case study as it exhibits highly variable and quickly evolving load patterns that depend on the concurrent activity of thousands of potential users. A given user has access only to a subset of the resources, as granted to her VO. When she submits jobs from her workstation, she uses an EGEE client known as a User Interface. A Workload Management Server receives and queues the jobs submitted before dispatching them to the connected computing centers. The gateway to each computing center is one or more Computing Elements which host a batch manager to distribute the workload over the center computing resources. Different batch systems are operated in different centers. Each site is configured independently with site-specific policies determining the number of queues available and queues maximal wall-clock times.

3.2 Latency measures

Latency measures were collected by submitting a very large number of probe jobs. These jobs, consisting in the execution of an almost null duration `/bin/hostname` command, are only impacted by the grid latency. In the remainder we assume that the job execution time is known and we only focus on the grid latency that can significantly vary between different runs of a same computation task. To avoid variations of the system load due to monitoring, a constant number of probes was maintained inside the system: a new probe was submitted each time another one completed. For each probe job, the job submission date, the job final status and the total duration were logged. The probe jobs were assigned a fixed 10,000 seconds timeout beyond which they were considered as outliers and canceled. This value is far greater than the average latency observed ($\approx$500 seconds).

Twelve trace sets collected at different times of the year and with different durations are exploited in this paper [7, 12]. They gather 10,893 probe jobs. A first trace acquired in September 2006 is denoted 2006-IX. The 11 other traces, acquired from the end of year 2007 to the beginning of year 2008 over one week each are denoted "year"-"week number". In the future, we plan to make more systematic experiments by exploiting logs of the Grid Observatory[3] that was recently set up. It gives practical foundations for the investigations conducted in this paper and it is a key to the systematic implementation of our methods in real applications.

3.3 Evaluation criteria

The submission strategies explored in the following are evaluated against two kinds of metric: the jobs performance (user point of view) and the number of submissions needed to achieve it (infrastructure point of view). The performance is assessed through the average latency time. In our probabilistic framework, an expression of the latency expectation (and its standard deviation) can be developed and then estimated from the traces mentioned above. This gives little insight when considering individual jobs but it makes perfect sense when considering applications involving a large number of jobs. The infrastructure load is assessed through the average number of jobs that is needed to achieve a given performance. Increasing the number of jobs in the system may impact the latency at some point. In our framework, we assume that the additional jobs have no measurable impact on the grid workload given its size and the number of jobs processed ($\geq$ 150 Kjobs / day on EGEE). As it will be demonstrated, this assumption makes sense as it is possible to achieve significant performance improvements for a small average number of additional jobs.

4 Single resubmission

When facing high and highly variable latencies, a simple resubmission strategy consists in waiting until a timeout value t_∞ and then canceling the job and resubmitting it, iterating this strategy until one job completes before t_∞. The modeling of the total latency J, including resubmissions, has already been studied in [9] and its expectation can be expressed as a function of individual jobs latency (R) and the timeout value (t_∞):

$$\boxed{E_J(t_\infty) = \frac{1}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty} (1 - \tilde{F}_R(u))du} \quad (1)$$

Minimizing this equation with respect to t_∞ gives the optimal timeout value (see figure 2).

[3]http://www.grid-observatory.org/

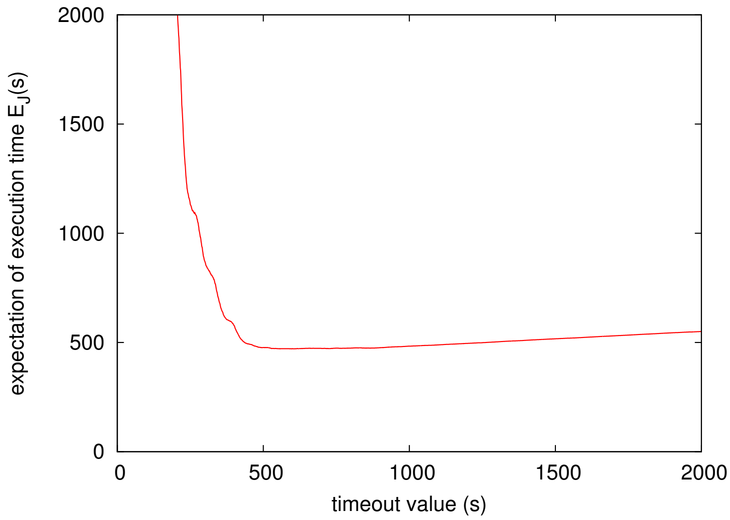

Figure 2: Expectation of execution time with respect to the timeout value. The minimal value of E_J gives the optimal timeout value.

The standard deviation σ_J is computed using the fact that $\sigma^2(X) = E(X^2) - E(X)^2$ (see appendix B for computation details):

$$\boxed{\begin{aligned}\sigma_J^2(t_\infty) \;=\; & -\frac{1}{\tilde{F}_R^2(t_\infty)}\left(\int_0^{t_\infty}(1-\tilde{F}_R(u))du\right)^2 \\ & +\frac{2}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty}u(1-\tilde{F}_R(u))du \\ & +\frac{2t_\infty(1-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^2}\int_0^{t_\infty}(1-\tilde{F}_R(u))du\end{aligned}} \qquad (2)$$

In table 1, different computations of means and standard deviations corresponding to the reference data presented in section 3 are displayed. We observe that the expected latency including resubmissions is of the same order of magnitude as the mean of all latencies smaller than 10,000 seconds (*i.e.* without outliers). For comparison, a low bound of the mean latency was computed, assuming that latencies greater than 10,000s were equal to 10,000s. This submission strategy allows to have a total latency in the same order of magnitude as if there were no outliers. On the other hand, we can observe that the standard deviation of latency including resubmission (σ_J) is smaller than the standard deviation of latencies smaller than 10,000s, except for one set of data where the value is almost similar (2008-01). It shows that for most periods this strategy reduces both the variability of the latency and the impact of outliers.

week number	mean $< 10^5$s	mean with 10^5s	E_J	σ_R $< 10^5$s	σ_J	$\Delta\sigma$
2006-IX	570s	1042s	471s	886s	331s	-63%
2007/08	469s	2089s	500s	723s	358s	-51%
2007-36	446s	2739s	510s	748s	370s	-51%
2007-37	506s	3639s	617s	848s	486s	-43%
2007-38	447s	2739s	531s	682s	399s	-42%
2007-39	489s	3533s	596s	741s	482s	-35%
2007-50	660s	2341s	628s	1046s	475s	-55%
2007-51	478s	1716s	517s	510s	353s	-31%
2007-52	443s	1685s	476s	582s	334s	-43%
2007-53	449s	1977s	482s	678s	330s	-51%
2008-01	434s	1678s	499s	317s	339s	+07%
2008-02	418s	1568s	441s	547s	278s	-49%
2008-03	538s	1484s	419s	1196s	269s	-78%

Table 1: Mean and standard variation of latency (R) and latency including resubmissions (J). The column "mean $< 10^5$" corresponds to the mean of latencies lower than 10,000 seconds. The column "mean with 10^5" is a low bound of the actual latency mean considering that latencies greater than 10,000s are equal 10,000s.

5 Multiple submissions

In order to further improve performance and to reduce chance of failure, multiple submission is often considered. A multiple submission strategy can easily be implemented within the EGEE Workload Management System (WMS) through burst submissions: for each job to be executed, a collection of b copies of this job is submitted. As soon as one job from the collection is running, all the other ones are canceled. If none of the jobs starts executing before the timeout value (t_∞), the whole collection is canceled and resubmitted.

We are now interested in the minimal execution time of the b parallel submissions. We assume that the laws of the jobs submitted in parallel are independent and that the probability for a job to finish before t is given by $\tilde{F_R}(t)$. Thus, the probability of all the b jobs to finish after t is given by $(1 - \tilde{F_R}(t))^b$. The probability of having at least one job running before t is thus given by $1 - (1 - \tilde{F_R}(t))^b$.

The new expected execution time can then be computed from equation 1 by replacing $\tilde{F_R}$ by $1 - (1 - \tilde{F_R}(t))^b$:

$$\boxed{E_J(t_\infty) = \frac{1}{1 - (1 - \tilde{F_R}(t_\infty))^b} \int_0^{t_\infty} (1 - \tilde{F_R}(u))^b du} \quad (3)$$

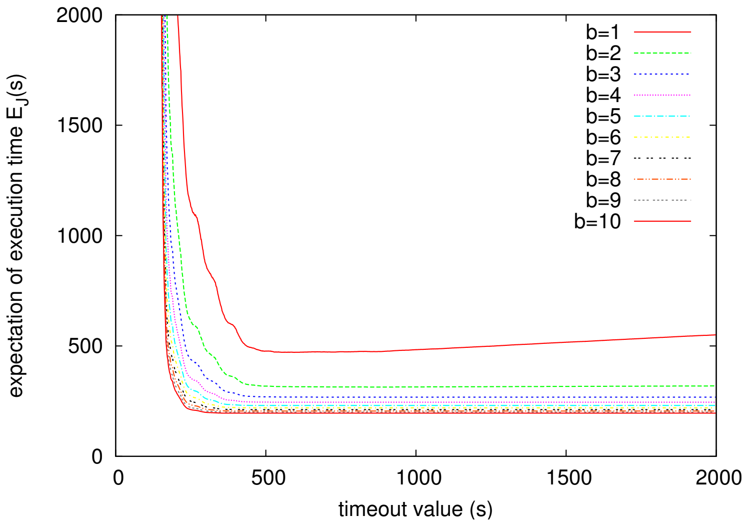

Figure 3: Expectation of execution time for different collection of b jobs, with respect to the timeout value.

and the standard deviation:

$$\boxed{\begin{aligned}\sigma_J^2(t_\infty) &= \frac{2}{1-(1-\tilde{F}_R(t_\infty))^b}\int_0^{t_\infty} u(1-\tilde{F}_R(u))^b du \\ &+\frac{2t_\infty(1-\tilde{F}_R(t_\infty))^b}{(1-(1-\tilde{F}_R(t_\infty))^b)^2}\int_0^{t_\infty}(1-\tilde{F}_R(u))^b du \\ &-\frac{1}{(1-(1-\tilde{F}_R(t_\infty))^b)^2}\left(\int_0^{t_\infty}(1-\tilde{F}_R(u))^b du\right)^2\end{aligned}} \quad (4)$$

Figure 3 plots the profile of E_J (from equation 3) for different values of b, using data *2006-IX*. As expected, the higher the value of b, the smaller the minimal expectation of the job execution time. We also observe that the slope of E_J after its minimal value is decreasing with b, leading to variations in the optimal t_∞ values. The optimal values for t_∞ and the minimal values of E_J for values of b from 1 to 20 are displayed in table 2. In the second group of columns, E_J variations are compared with the case $b = 1$ (which corresponds to the single resubmission strategy). Very significant performance improvements are obtained, even for low values of b: for only 5 redundant submissions, E_J already drops by a factor 2. Moreover, the standard deviation σ_J is also decreasing, concentrating the values of J around E_J. Yet, these decrease slow down when b increases further. In the third group of columns we compare results obtained for a given value of b to the ones obtained for $b-1$ in order to measure the improvement of E_J with one unit of b. E_J is decreasing, significantly faster for smaller values of b than larger ones. This result is intuitive: adding one job to the collection has much more impact if the collection is very small than if it already contains many jobs.

b	opt. t_∞	best E_J	σ_J	ΔE_J /(b=1)	Δb /(b=1)	ΔE_J /(b-1)	Δb /(b-1)
1	596s	471s	331s				
2	880s	314s	148s	-33%	200%	-33.4%	100%
3	881s	268s	92s	-43%	300%	-14.6%	50%
4	881s	245s	73s	-48%	400%	-8.6%	33.3%
5	887s	230s	63s	-51%	500%	-6.0%	25%
6	1071s	220s	57s	-53%	600%	-4.6%	20%
7	1071s	212s	51s	-55%	700%	-3.7%	16.7%
8	1071s	205s	47s	-57%	800%	-3.0%	14.3%
9	1071s	200s	43s	-58%	900%	-2.6%	12.5%
10	1247s	196s	40s	-59%	1000%	-2.2%	11.1%
11	1247s	192s	38s	-59%	1100%	-1.9%	10%
12	1247s	189s	35s	-60%	1200%	-1.6%	9.1%
13	2643s	186s	33s	-61%	1300%	-1.4%	8.3%
14	1740s	184s	32s	-61%	1400%	-1.3%	7.7%
15	1199s	182s	30s	-62%	1500%	-1.1%	7.1%
16	980s	180s	29s	-62%	1600%	-1.0%	6.7%
17	853s	178s	27s	-62%	1700%	-0.9%	6.3%
18	792s	177s	26s	-63%	1800%	-0.9%	5.9%
19	730s	175s	25s	-63%	1900%	-0.8%	5.6%
20	688s	174s	24s	-63%	2000%	-0.7%	5.3%

Table 2: Different values of the number of jobs in the collection (b) leads to different values of optimal timeout and best expectation of execution time. A significant speed-up is achieved by the multi-submission strategy, even for low values of b.

In figure 4, the optimal values of E_J and associated standard-deviation σ_J are plotted for different periods of time with respect to the number of jobs in parallel. The decreasing curves confirm the previous observations.

6 Delayed resubmission strategy

The multiple submission strategy is efficient but aggressive for the infrastructure. An alternate delayed resubmission strategy, derived from the single resubmission is presented here. As illustrated in figure 5, it consists in submitting a single job, waiting until t_0 and then, if it is not running yet, launching a copy of this job without canceling the first one before t_∞, and iterating this process until one job is running.

In order not to have more than 2 identical jobs in the system at the same time, we impose $0 < t_0 < t_\infty$ and $(t_\infty - t_0) < t_0$ (this ensures that job 1 is canceled before job 3 is submitted, as illustrated on figure 5). The probability for a single job to timeout is given by $q = 1 - \tilde{F}_R(t_\infty)$. If a job starts running at time t in the interval $[nt_0,\ (n-1)t_0+t_\infty]$ (I_0 on figure 5), this means that exactly $(n-1)$ jobs have timed-out (probability q^{n-1}) and that either latency of job n is between t_0 and $(t-(n-1)t_0)$ (probability $(\tilde{F}_R(t-(n-1)t_0) - \tilde{F}_R(t_0))$) or latency of job $n+1$ is lower than $(t-nt_0)$ (probability $\tilde{F}_R(t-nt_0)$). Since these last two events may both occur, the probability that at least one of them occurs is equal to the probability of their union minus the probability of their intersection, i.e.:

$$\begin{aligned} F_J(t) &= P(J < t | t \in [nt_0,\ (n-1)t_0 + t_\infty]) \\ &= P(J < nt_0) + (1 - \tilde{F}_R(t_\infty))^{n-1} \\ &\quad . \Big(\tilde{F}_R(t-(n-1)t_0) - \tilde{F}_R(t_0) + \tilde{F}_R(t - nt_0) \\ &\quad - (\tilde{F}_R(t-(n-1)t_0) - \tilde{F}_R(t_0))\tilde{F}_R(t - nt_0) \Big) \end{aligned}$$

Otherwise, if a job starts running at time t in the interval $[(n-1)t_0 + t_\infty,\ (n+1)t_0]$ (I_1 on figure 5), this means that exactly n jobs have timed out (with probability q^n) and that the latency of job $(n+1)$ is lower than $(t - nt_0)$ (probability $\tilde{F}_R(t-nt_0)$).

$$\begin{aligned} F_J(t) &= P(J < t | t \in [(n-1)t_0 + t_\infty,\ (n+1)t_0]) \\ &= P(J < (n-1)t_0 + t_\infty) + (1 - \tilde{F}_R(t_\infty))^n \tilde{F}_R(t - nt_0) \end{aligned}$$

Deriving these last 2 equations leads to :

$$\begin{cases} \forall t \in [nt_0,\ (n-1)t_0 + t_\infty] \\ \qquad f_J(t) = q^{n-1} \Big(\tilde{f}_R(t-(n-1)t_0) \\ \qquad + (1 + \tilde{F}_R(t_0))\tilde{f}_R(t - nt_0) \\ \qquad - \tilde{f}_R(t-(n-1)t_0).\tilde{f}_R(t-nt_0) \Big) \\ \forall t \in [(n-1)t_0 + t_\infty,\ (n+1)t_0] f_J(t) = q^n \tilde{f}_R(t - nt_0) \end{cases}$$

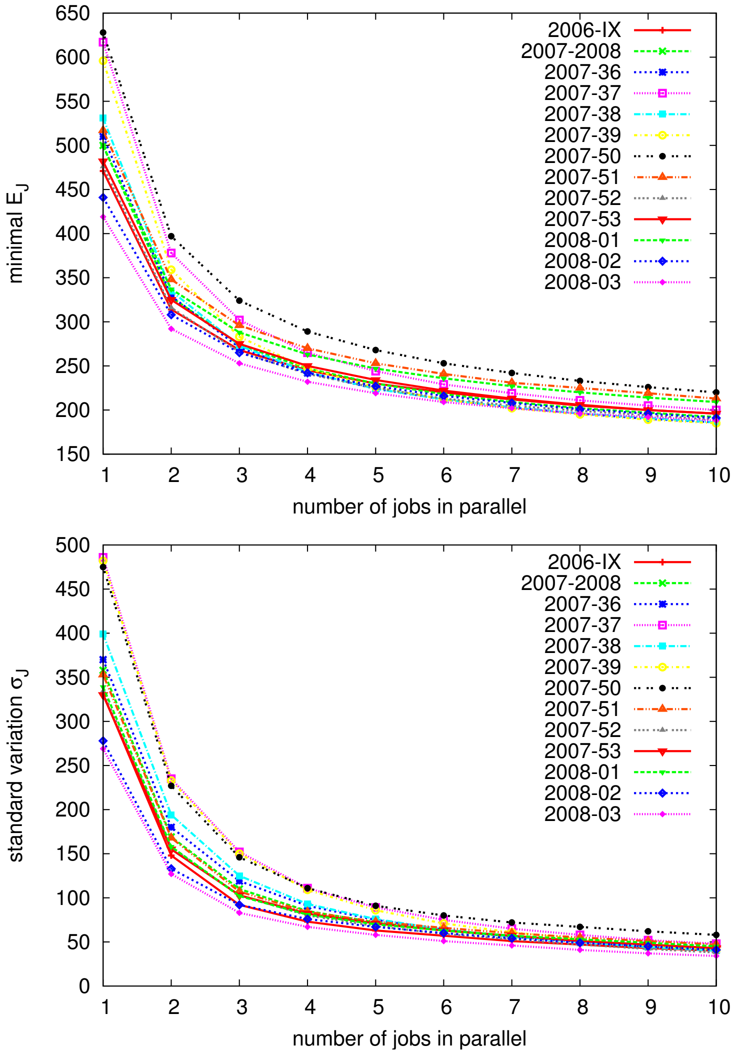

Figure 4: Evolution of minimal values of E_J and the associated σ_J values with respect to the number of jobs in the multi-submission (b). Each curve corresponds to a set of data.

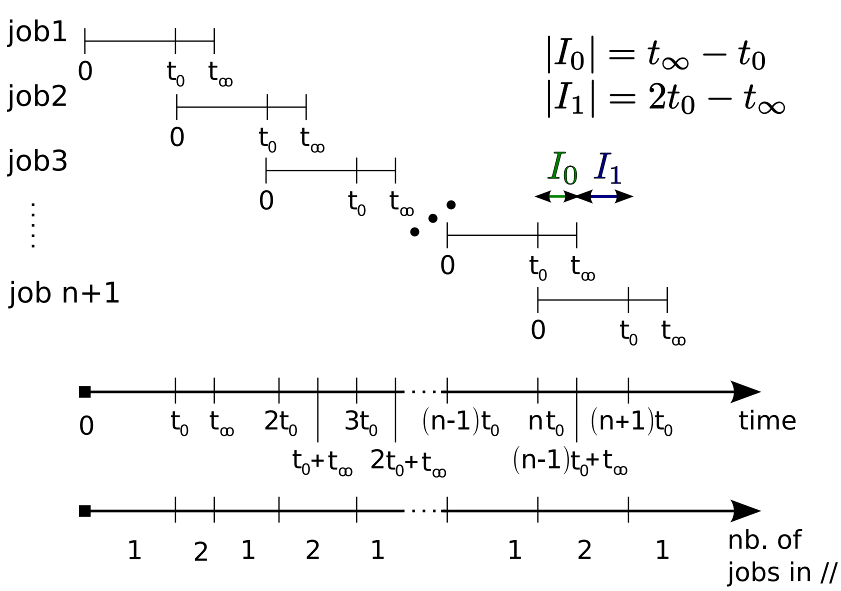

Figure 5: Principle of the delayed resubmission strategy: a single job is first submitted. At t_0, if the job has not started, a copy of this job is submitted. If the first job is still not completed at t_∞, it is canceled. This strategy is iterated until a job is completed.

Finally, by integration, and by replacing the integration variable t with $u = t - nt_0$ or $v = t - (n-1)t_0$, we get:

$$
\begin{aligned}
E_J &= \int_0^\infty t f_J(t)dt \\
&= \int_0^{t_0} t f_J(t)dt \\
&+ \sum_{n=1}^{\infty} \left(\int_{nt_0}^{(n-1)t_0+t_\infty} t f_J(t)dt + \int_{(n-1)t_0+t_\infty}^{(n+1)t_0} t f_J(t)dt \right) \\
&= \int_0^{t_0} t \tilde{f}_R(t)dt + \sum_{n=1}^{\infty} q^{n-1} \int_{t_0}^{t_\infty} (v + (n-1)t_0)\tilde{f}_R(v)dv \\
&+ (1 + \tilde{F}_R(t_0)) \sum_{n=1}^{\infty} q^{n-1} \int_0^{t_\infty - t_0} (u + nt_0)\tilde{f}_R(u)du \\
&+ \sum_{n=1}^{\infty} q^n \int_{t_\infty - t_0}^{t_0} (u + nt_0)\tilde{f}_R(u)du \\
&- \sum_{n=1}^{\infty} q^{n-1} \int_0^{t_\infty - t_0} (u + nt_0)\tilde{f}_R(u + t_0).\tilde{f}_R(u)du
\end{aligned}
$$

Grouping terms and replacing the integer series by their values leads to:

$$
\begin{aligned}
E_J &= \int_0^{t_0} t\tilde{f}_R(t)dt \\
&+ \frac{1}{1-q}\int_{t_0}^{t_\infty} v\tilde{f}_R(v)dv + \frac{qt_0}{(1-q)^2}(\tilde{F}_R(t_\infty) - \tilde{F}_R(t_0)) \\
&+ \frac{1+\tilde{F}_R(t_0)}{1-q}\int_0^{t_\infty - t_0} u\tilde{f}_R(u)du \\
&+ \frac{t_0(1+\tilde{F}_R(t_0))}{(1-q)^2}\tilde{F}_R(t_\infty - t_0) + \frac{q}{1-q}\int_{t_\infty - t_0}^{t_0} u\tilde{f}_R(u)du \\
&+ \frac{qt_0}{(1-q)^2}(\tilde{F}_R(t_0) - \tilde{F}_R(t_\infty - t_0)) \\
&- \frac{t_0}{(1-q)^2}\int_0^{t_\infty - t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- \frac{1}{1-q}\int_0^{t_\infty - t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

And finally (more details are given in appendix C):

$$
\boxed{
\begin{aligned}
E_J(t_0,t_\infty) &= \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty} u\tilde{f}_R(u)du \\
&+ \frac{\tilde{F}_R(t_0)}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty - t_0} u\tilde{f}_R(u)du + \frac{t_0}{\tilde{F}_R(t_\infty)} \\
&+ t_0\frac{\tilde{F}_R(t_\infty - t_0)}{\tilde{F}_R(t_\infty)} + t_0\frac{\tilde{F}_R(t_0)\tilde{F}_R(t_\infty - t_0)}{\tilde{F}_R^{\,2}(t_\infty)} \\
&- t_0 + \int_0^{t_\infty - t_0} u\tilde{f}_R(u)du \\
&- \frac{t_0}{\tilde{F}_R(t_\infty)^2}\int_0^{t_\infty - t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty - t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
}
\tag{5}
$$

This expression has to be minimized numerically with respect to t_0 and t_∞. In figure 6 the surface profile of E_J has been computed using dataset *2006-IX*. This dataset leads to a best t_0 value of 339s, a best t_∞ value of 485s and a minimum value for the expected execution time of 431s, which is smaller than the single resubmission strategy but higher than the multiple resubmission strategy for $b \geq 2$. Although minimizing E_J leads to the best performance from a user point of view, it might also load the infrastructure by increasing the number of redundant jobs, which we study in the following.

6.1 Number of parallel jobs

With the delayed resubmission strategy, a variable number of jobs may be running on the infrastructure at any time. Let $N_{//}$ denote the average number of jobs needed,

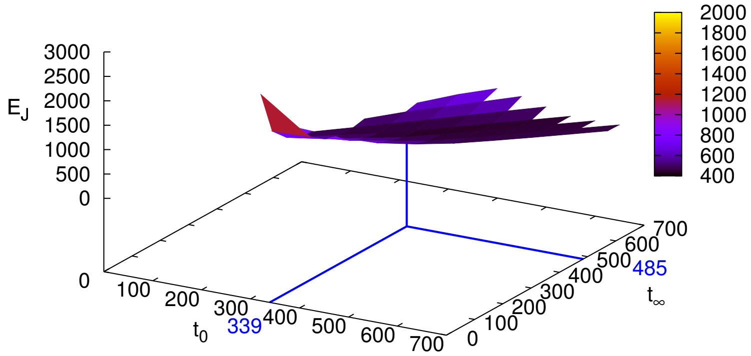

Figure 6: Expectation of execution time with respect to t_0 and t_∞, in the case of delayed resubmission strategy. The surface presents a minimum ($t_0 = 339s$, $t_\infty = 485s$ and $E_J = 431s$).

computed as the normalized sum of the number of parallel jobs running at each instant. $N_{//}$ is to be compared with the value of b in the case of multiple resubmissions. Let l be the latency of a job with parameters t_0 and t_∞. We can notice that, after the first period $[0; t_0]$, the number of jobs in parallel is periodic with a t_0-period of time. In the following, let n be an integer such that l is in $[nt_0, (n+1)t_0[$.

Three cases have to be distinguished:

$\mathbf{n = 0}$; Then, obviously, $N_{//} = 1$.

$\mathbf{n = 1}$; Two different cases have to be considered:

- if $\mathbf{l < t_\infty}$: only one job is running until t_0. Then, a second job is running in parallel with the first one during a period of $(l - t_0)$ which leads to :

$$N_{//} = \frac{t_0 + 2(l - t_0)}{l} = 2 - \frac{t_0}{l}$$

- if $\mathbf{l \geq t_\infty}$: only one job is running until t_0. Then, two jobs are running in parallel during a period of $(t_\infty - t_0)$. After that, the first job is canceled and there will be only one job running during $(l - t_\infty)$ leading to :

$$N_{//} = \frac{t_0 + 2(t_\infty - t_0) + (l - t_\infty)}{l}$$

$\mathbf{n > 1}$; For both cases detailed below, only one job is running until t_0. Then, for $(n-1)$ period of t_0 time, we will have two jobs in parallel during $|I_0| = (t_\infty - t_0)$ and only one job during $|I_1| = (2t_0 - t_\infty)$.

- if $\mathbf{l} \in [\mathbf{nt_0}; (\mathbf{n}-\mathbf{1})\mathbf{t_0} + \mathbf{t}_\infty[$ (I_0 on figure 5): after the $(n-1)t_0$ first periods, two jobs are running in parallel during $(l - nt_0)$ which leads to :

$$N_{//} = \frac{t_0 + (n-1)t_\infty + 2(l - nt_0)}{l}$$

- if $\mathbf{l} \in [(\mathbf{n}-\mathbf{1})\mathbf{t_0} + \mathbf{t}_\infty; (\mathbf{n}+\mathbf{1})\mathbf{t_0}[$ (I_1 on figure 5): after the $(n-1)t_0$ first periods, two jobs are running in parallel during $|I_0| = (t_\infty - t_0)$ and then only one job during $(l - ((n-1)t_0 + t_\infty))$ which leads to :

$$N_{//} = \frac{t_0 + (n-1)t_\infty + 2(t_\infty - t_0) + (l - (n-1)t_0 - t_\infty)}{l}$$

It can be verified that $\lim_{n\to\infty} N_{//} = \frac{t_\infty}{t_0}$. However, considering realistic values of l and t_∞, n will usually be small. We can demonstrate that $N_{//} \in [1; 2 - \frac{1}{n+1}]$, thus leading to a maximum of 1.5 in the case of $n = 1$ (see details in appendix E). In the following, we will experiment different values of $N_{//}$ in order to study the variations of the minimal value of l.

6.2 Imposing the ratio $\frac{t_\infty}{t_0}$.

Although this ratio can easily be imposed, it only corresponds to the asymptotic behavior of $N_{//}$ and not to $N_{//}$'s actual value. $N_{//}$ is thus computed in each case, using the minimal E_J values computed using equation 5.

In table 3, the results of the minimization of equation 5 for different values of the ratio $\frac{t_\infty}{t_0}$ are presented, leading to different values of $N_{//}$. The minimal values of E_J correspond either to $n = 0$ ($E_J < t_0$), where $N_{//} = 1$, or to $n = 1$ ($N_{//}$ between 1 and 1.45).

The minimal values of E_J are compared with the number of jobs in parallel on figure 7. The minimal value for the delayed resubmission strategy, obtained from a global minimization of equation 5, is $E_J = 431s$ for a mean of 1.2 jobs in parallel. This minimal value is lower than the one obtained with the single resubmission strategy by 8.3%. However, we obtain a lower value with the multiple submission strategy with at least two jobs in parallel. To fairly compare different strategies, a measure of the cost induced by an increase of the mean number of jobs in parallel is needed.

7 Discussion on the strategies cost

Although submitting the same job twice increases the grid load, it still leads to a global benefit for the infrastructure if the gain in time is higher than 2. Indeed, in this case, the expectation of the number of jobs in the system decreases, as illustrated on figure 8. This idea can be extended to any number of jobs, trying to satisfy this relation:

$$E_J(\text{ delayed sub. with } N_{//}) < \frac{E_J(\text{ single resub. with } b = 1)}{N_{//}}$$

$\frac{t_\infty}{t_0}$	$N_{//}$	best t_∞	best t_0	min E_J	$\Delta(100\%)$ 471s
1.1	1	556s	505s	458s	-2.7%
1.2	1	556s	463s	447s	-5.0%
1.3	1.07	528s	406s	438s	-6.9%
1.4	1.18	496s	354s	432s	-8.2%
1.5	1.32	445s	297s	434s	-7.7%
1.6	1.37	435s	272s	444s	-5.6%
1.7	1.39	431s	254s	457s	-2.9%
1.8	1.41	426s	237s	462s	-1.9%
1.9	1.47	425s	224s	466s	-1%
2	1.45	423s	211s	469s	-0.5%

Table 3: Delayed resubmission strategy: for each ratio $\frac{t_\infty}{t_0}$, the minimal E_J is computed. All E_J values are below E_J from the single resubmission strategy (471s). These results are computed on the *2006-IX* dataset.

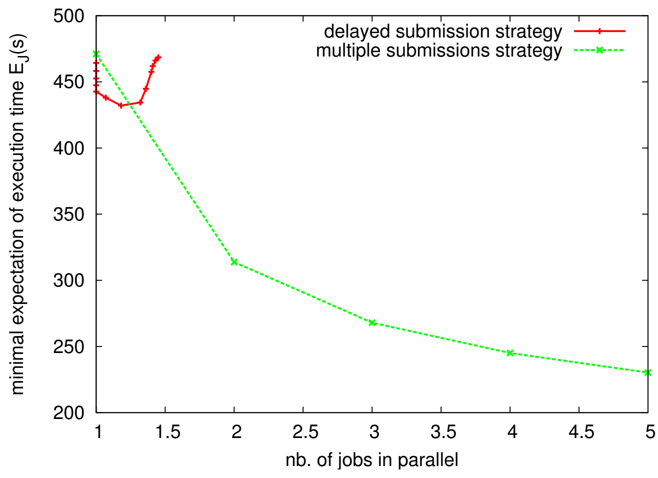

Figure 7: Minimal expected execution time according to delayed resubmission strategy (plain curve) or multiple submission strategy (dashed curve), with respect to the mean number of job copies running in parallel.

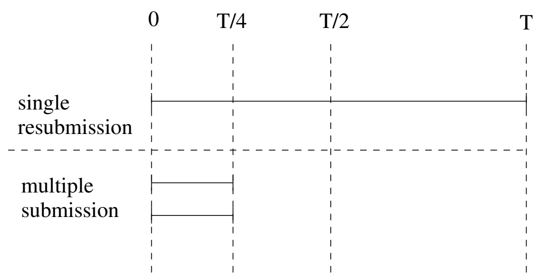

Figure 8: Multiple submission strategies can also reduce the global load of the infrastructure when they yield sufficient time gain. Top: the single resubmission strategy leads to an average number of jobs of 1 on [0,T]. Bottom: a multiple submission strategy reduces it to 0.5 on this time period.

The cost of the strategy is then:

$$\boxed{\Delta_{cost} = N_{//} * \frac{E_J(\text{with} N_{//})}{E_J(\text{ with } b = 1)}} \quad (6)$$

for different job numbers.

Using this definition, the cost of the single resubmission strategy detailed in Section 4 is 1. In table 4, different samples of $N_{//}$, E_J and Δ_{cost} are displayed for the different strategies studied in this paper. Obviously, in the multiple submission strategy, we have $N_{//} = b$. Figure 9 presents the values of Δ_{cost} with respect to $N_{//}$. For integer values of $N_{//}$ (corresponding to the multiple submission strategy), values of Δ_{cost} values are increasing and greater than 1. For values of $N_{//}$ smaller than 2, the values of Δ_{cost}, starting from 1, are decreasing until a minimal value of 0.94, and then increasing beyond 1. All values of Δ_{cost} smaller than 1 indicate that the load on the grid is lower than a single running job (the strategy of single resubmission). Using the smallest value of Δ_{cost} leads to the smallest occupation of the grid while reducing the latency. The results that are displayed in table 4 are those, given a value of $\frac{t_\infty}{t_0}$, that minimize E_J. Considering the minimization with respect to the Δ_{cost} value, the minimum is reached for $\Delta_{cost} = 0.93$, $t_0 = 439s$ and $t_\infty = 579s$ leading to $E_J = 439s$ (less than $E_{J(b=1)} = 471$s). Moreover, a consequence of reducing the grid load might be a reduction of the latencies. This has not been studied in this paper and it is subject for future work.

7.1 Results on data from 2007-2008

The left side of table 5 presents, for each week of the dataset from 2007-2008, the minimum value of Δ_{cost} obtained using the delayed resubmission strategy. For the first 5 weeks, the minimal value of Δ_{cost} is higher than 1 while, for the other 6 weeks, including the whole period, Δ_{cost} presents a minimum less than 1, as it was the case for the dataset studied in the previous section. This shows that, depending on the grid workload, the

$N_{//}$	$\frac{t_\infty}{t_0}$	min E_J	Δ_{cost}	$N_{//}$	min E_J	Δ_{cost}
1		471s	1	2	314s	1.3
1	1.1	458s	0.97	3	268s	1.7
1	1.15	453s	0.96	4	245s	2.1
1	1.2	447s	0.95	5	230s	2.4
1	1.25	443s	$\boxed{0.94}$	6	220s	2.8
1.07	1.3	438s	1.00	7	212s	3.1
1.18	1.4	432s	1.09	8	205s	3.5
1.32	1.5	434s	1.22	9	200s	3.8
1.36	1.6	445s	1.29	10	196s	4.2
1.40	1.7	458s	1.36	20	174s	7.4
1.41	1.8	462s	1.38	40	161s	14
1.43	1.9	466s	1.42	60	156s	20
1.45	2.0	469s	1.44	80	154s	26
				100	152s	32

Table 4: In the case of the delayed resubmission strategy, for each $\frac{t_\infty}{t_0}$ value, the minimal E_J is computed. Corresponding values of $N_{//}$ and Δ_{cost} are thus given. We observe that a ratio $\frac{t_\infty}{t_0}$ of 1.25 appears to be the optimal solution with respect to the Δ_{cost} value. For other strategies, minimal E_J is computed from $N_{//}$. For a higher number of jobs, the cost increases. These results have been computed on the 2006-IX dataset.

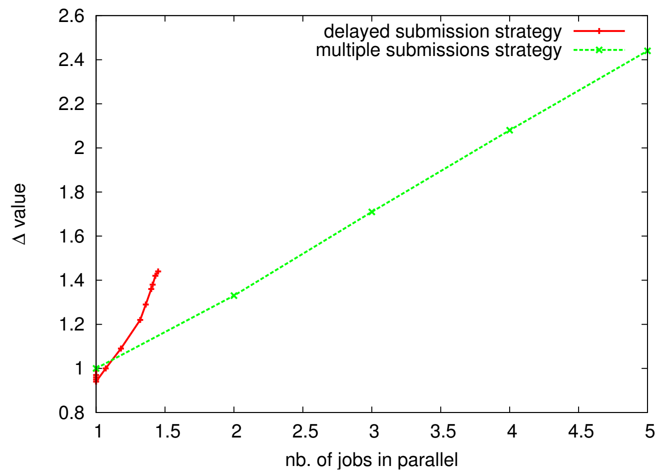

Figure 9: Δ value according to 2 different strategies: delayed resubmission strategy or multiple submission strategy, with respect to the mean number of job copies running in parallel.

delayed strategy is not always optimal in term of Δ_{cost}: the single submission should to be chosen instead.

For each period of time, the values of t_0 and t_∞ corresponding to the minimal value of Δ_{cost} are given. We have also tested the stability of such minimal values by adding variations of up to $\pm$ 5 seconds to each of t_0 and t_∞. The study was limited to integer values of t_0 and t_∞ because having higher precision of resubmission is not realistic in practice. The maximal values of Δ_{cost} and the maximum relative difference with respect to the minimum are given on the right of table 5 for all minima less than 1. Some Δ_{cost} stay below 1 while others grow more than 1 but still below 1.1 where the highest relative difference is of 14%. This shows a relative stability that needs to be enforced by a good estimation of both optimals t_0 and t_∞.

7.2 Practical implementation

Up to now, traces were studied a posteriori. However, exploiting the Δ_{cost} optimization strategy in practice requires to collect data for estimating t_0 and t_∞ prior to the jobs execution. In the experiment reported in table 6, the variations of Δ_{cost} for the different values of t_0 and t_∞ that have been obtained on each time period are studied. Assuming that Δ_{cost} was computed from the measurements collected during any of the periods randomly considered, the value of Δ_{cost} is shown (the optimal value, corresponding to the studied period's measurements, is underlined). These results show a maximal variation of 13% (mean variation of 9%). Assuming now that Δ_{cost} was computed from the measurements collected the week preceeding, the last column in the table displays the variation of Δ_{cost} between the current and the preceeding week. In this case, the relative difference is never larger than 6% and when Δ_{cost} is higher than 1, it is precised to 10^{-3}.

8 Conclusion

In this paper we have studied 3 different job submission strategies that users can adopt in order to reduce the latency they experience on production grids. For each of these strategies, we have established a model of the expectation of the total latency (including resubmission) and its standard deviation. We have shown that all these strategies reduce the latency. Moreover, the delayed resubmission strategy reduces the latency requiring less than 2 jobs in parallel while the multiple submission strategy gives the highest latency reduction but at a higher cost.

We have proposed a cost criterion which characterizes the conditions under which it is possible to obtain both a latency smaller than in the case of single resubmission and fewer parallel jobs.

Future work will concern the validation of this approach with real applications. The impact of each strategy on grid-applications makespan can be measured and averaged to take into account evolving experimental conditions. In a second step, the impact of

week	opt. t_0	opt. t_∞	opt. Δ_{cost}	E_J	max Δ_{cost}	max $\Delta\%$
2007-36	422	423	1.001	510		
2007-37	421	422	1.000	616		
2007-38	427	428	1.001	530		
2007-39	435	436	1.001	595		
2007-50	466	467	1.001	627		
2007-51	499	662	0.954	494	1.09	14 %
2007-52	455	595	0.955	455	0.97	1.2 %
2007-53	463	613	0.961	463	0.97	1.4 %
2008-01	489	525	0.981	489	1.03	4.7 %
2008-02	420	575	0.953	420	1.09	14 %
2008-03	395	530	0.943	395	0.95	1.3 %
2007/08	481	635	0.963	481	1.09	13 %

Table 5: Minimal Δ_{cost} values for the different periods with corresponding values of t_0, t_∞ and E_J. All E_J values are below the ones obtained with the single resubmission strategy (see table 1). For the cases where the minimum is lower than 1.0, variations of Δ_{cost} around optimal t_0 and t_∞ values (radius 5): maximal value and relative difference with the maximal value.

all grid users exploiting the same strategy can be simulated in a controlled environment.

Acknowledgments

The authors would like to thank Pierre Bernhard and Roger Marlin for fruitful discussions. This work is partially funded by the French national research agency (ANR), NeuroLog project[4] under contract number ANR-06-TLOG-024 and CNRS ST2I PEPS "Grid Observatory". We are grateful to the EGEE European project for providing the grid infrastructure and user assistance.

References

[1] J. Andreeva, S. Campana, F. Fanzago, and J. Herrala. High-Energy Physics on the Grid: the ATLAS and CMS Experience. *Journal of Grid Computing (JGC)*, 6(1):3–13, Mar. 2008.

[2] G. Aparicio, I. Blanquer Espert, and V. Hernández García. A Highly Optimized Grid Deployment: the Metagenomic Analysis Example. In *Global Healthgrid: e-*

[4] Neurolog: `http://neurolog.polytech.unice.fr`

week	t_0	t_∞	E_J	Δ_{cost}	Max diff	Diff /prev
2007-51	499	662	494	$\underline{0.954}$	13%	–
	455	595	483	0.987		
	463	613	485	0.980		
	489	525	509	1.023		
	420	575	479	1.040		
	395	530	477	1.083		
2007-52	499	662	482	1.012	8%	6%
	455	595	455	$\underline{0.955}$		
	463	613	457	0.960		
	489	525	473	0.994		
	420	575	447	0.996		
	395	530	443	1.031		
	481	635	462	0.970		
2007-53	499	662	494	1.025	9%	1%
	455	595	461	0.971		
	463	613	463	$\underline{0.961}$		
	489	525	478	0.993		
	420	575	454	1.015		
	395	530	449	1.046		
	481	635	468	0.972		
2008-01	499	662	505	1.025	7%	2%
	455	595	479	1.007		
	463	613	481	1.001		
	489	525	489	$\underline{0.981}$		
	420	575	468	1.034		
	395	530	460	1.053		
	481	635	487	0.987		
2008-02	499	662	458	1.040	9%	5%
	455	595	425	0.963		
	463	613	426	0.967		
	489	525	439	0.996		
	420	575	420	$\underline{0.953}$		
	395	530	416	0.990		
	481	635	430	0.976		
2008-03	499	662	432	1.030	9%	1%
	455	595	401	0.957		
	463	613	401	0.958		
	489	525	416	0.993		
	420	575	399	0.952		
	395	530	395	$\underline{0.943}$		
	481	635	405	0.965		
2007 / 2008	499	662	514	1.058	9.8%	
	455	595	474	0.988		
	463	613	476	0.978		
	489	525	496	1.008		
	420	575	469	1.038		
	395	530	461	1.056		
	481	635	481	$\underline{0.963}$		

Table 6: Variations of E_J and Δ_{cost} for different values of t_0 and t_∞ corresponding to optimal Δ_{cost} values for each week and the whole period 2007/08.

Science Meets Biomedical INformatics (Healthgrid'08), pages 105 –115, Chicago, USA, May 2008. Healthgrid, IOS Press.

[3] H. Casanova. Benefits and Drawbacks of Redundant Batch Requests. *Journal of Grid Computing (JGC)*, 2(5):888–903, 2007.

[4] K. Christodoulopoulos, V. Gkamas, and E. A. Varvarigos. Statistical Analysis and Modeling of Jobs in a Grid Environment. *Journal of Grid Computing (JGC)*, 6(1):77-101, Mar. 2008.

[5] D. Feitelson. *Workload modeling for performance evaluation*, pages 114–141. LNCS vol 2459, Sept. 2002.

[6] C. Germain, C. Loomis, J. T. Mościcki, and R. Texier. Scheduling for Responsive Grids. *Journal of Grid Computing (JGC)*, 6(1):15–27, Mar. 2008.

[7] T. Glatard, D. Lingrand, J. Montagnat, and M. Riveill. Impact of the execution context on Grid job performances. In *International Workshop on Context-Awareness and Mobility in Grid Computing (WCAMG'07)*, pages 713–718, May 2007.

[8] D. Lingrand, J. Montagnat, and T. Glatard Modeling user submission strategies on production grids In *International Symposium on High Performance Distributed Computing (HPDC'09)*, Munich, Germany, 11-13 June 2009.

[9] T. Glatard, J. Montagnat, and X. Pennec. Optimizing jobs timeouts on clusters and production grids. In *CCGrid'07*, pages 100–107, Rio de Janeiro, May 2007.

[10] N. Jacq, V. Breton, H.-Y. Chen, L.-Y. Ho, M. Hofmann, V. Kasam, H.-C. Lee, Y. Legré, S. C. Lin, A. Maaí, E. Medernach, I. Merelli, L. Milanesi, G. Rastelli, M. Reichstadt, J. Salzeman, H. Schwichtenberg, Y.-T. Wu, and M. Zimmermann. Virtual screening on large scale grids. *Parallel Computing*, 23(4-5):289–301, 2007.

[11] H. Li, D. Groep, and L. Walters. Workload Characteristics of a Multi-cluster Supercomputer. In *Job Scheduling Strategies for Parallel Processing*, pages 176–193. Springer Verlag, 2004.

[12] D. Lingrand, J. Montagnat, and T. Glatard. Estimation of latency on production grid over several weeks. In *ICT4Health*, Manila, Philippines, Feb. 2008.

[13] M. Niinimaki, X. Zhou, A. Depeursinge, A. Geissbuhler, and H. Muller. Building a Community Grid for Medical Image Analysis inside a Hospital, a Case Study. In S. Olabarriaga, D. Lingrand, and J. Montagnat, editors, *MICCAI-Grid Workshop*, pages 3–12, New York, NY, USA, Sept. 2008.

[14] M. J. Pitkanen, X. Zhou, A. E. Hyvarinen, and H. Muller. Using the Grid for Enhancing the Performance of a Medical Image Search Engine. In *21st IEEE International Symposium on Computer-Based Medical Systems (CBMS'08)*, pages 367–372, Jyvaskyla, Finland, June 2008.

[15] G. Sabin, R. Kettimuthu, A. Rajan, and P. Sadayappan. Scheduling of Parallel Jobs in a Heterogeneous Multi-Site Environment. In *JSSPP'03*, volume LNCS 2872, pages 87–104, 2003.

[16] J. Schopf and F. Berman. Stochastic Scheduling. In *Supercomputing (SC'99)*, Portland, USA, 1999.

[17] V. Subramani, R. Kettimuthu, S. Srinivasan, and P. Sadayappan. Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests. In *International Symposium on High Performance Distributed Computing (HPDC)*, pages 359–366, Edinburgh, Scotland, July 2002.

A Usefull formulae

A.1 Integer series

$$\frac{1}{(1-z)} = \sum_{n=0}^{\infty} z^n \tag{7}$$

$$\frac{1}{(1-z)^2} = \sum_{n=0}^{\infty}(n+1)z^n \Rightarrow \sum_{n=0}^{\infty} nz^n = \frac{z}{(1-z)^2} \tag{8}$$

$$\frac{1}{(1-z)^3} = \sum_{n=0}^{\infty} \frac{(n+1)(n+2)}{2} z^n \Rightarrow \sum_{n=0}^{\infty} n^2 z^n = \frac{z(z+1)}{(1-z)^3} \tag{9}$$

$$\sum_{n=1}^{\infty} z^{n-1} = \sum_{n=0}^{\infty} z^n = \frac{1}{1-z} \tag{10}$$

$$\sum_{n=1}^{\infty} z^n = \frac{1}{1-z} - 1 = \frac{z}{1-z} \tag{11}$$

$$\sum_{n=1}^{\infty}(n-1)z^{n-1} = \frac{z}{(1-z)^2} = \sum_{n=1}^{\infty} nz^{n-1} - \sum_{n=1}^{\infty} z^{n-1} \tag{12}$$

$$\sum_{n=1}^{\infty} nz^{n-1} = \frac{z}{(1-z)^2} + \frac{1}{1-z} = \frac{1}{(1-z)^2} \tag{13}$$

$$\sum_{n=1}^{\infty} n^2 z^{n-1} = \frac{z(z+1)}{(1-z)^3} - \frac{1}{1-z} + \frac{2}{(1-z)^2} = \frac{z+1}{(1-z)^3} \tag{14}$$

A.2 Integration by parts

$$\begin{aligned}\int_0^a uf'(u)du &= [uf(u)]_0^a - \int_0^a f(u)du \\ &= af(a) - \int_0^a f(u)du\end{aligned} \tag{15}$$

$$\begin{aligned}\int_0^a u^2 f'(u)du &= [u^2 f]_0^a - 2\int_0^a uf(u)du \\ &= a^2 f(a) - 2\sum_{k=0}^{a}\int_k^{k+1} uf(u)du \\ &= a^2 f(a) - 2\sum_{k=0}^{a}(k+1-k)kf(k)) \\ &= a^2 f(a) - 2\sum_{k=0}^{a} kf(k)\end{aligned} \tag{16}$$

$$\begin{aligned}\int_0^a ubf'(u)(1-f(u))^{b-1}du &= -\left[u(1-f(u))^b\right]_0^a + \int_0^a (1-f(u))^b du \\ &= -a(1-f(a))^b + \int_0^a (1-f(u))^b du\end{aligned} \tag{17}$$

$$\begin{aligned}\int_a^b u^2 f'(u)du &= [u^2 f(u)]_a^b - 2\int_a^b uf(u)du \\ &= b^2 f(b) - a^2 f(a) - 2\int_a^b u(f(u)-1)du - 2\int_a^b udu \\ &= b^2 f(b) - a^2 f(a) + 2\int_a^b u(f(u)-1)du - [u^2]_a^b \\ &= a^2(1-f(a)) - b^2(1-f(b)) + 2\int_a^b u(1-f(u))du\end{aligned} \tag{18}$$

$$\begin{aligned}\int_a^b uf'(u)du &= [uf(u)]_a^b - \int_a^b f(u)du \\ &= bf(b) - af(a) - \int_a^b (f(u)-1)du - \int_a^b 1du \\ &= a(1-f(a)) - b(1-f(b)) + \int_a^b (1-f(u))du\end{aligned} \tag{19}$$

B Computation of σ in the case of single resubmission.

By definition:

$$\text{var}(X) = E(X^2) - (E(X))^2 = \int_0^{\infty} t^2 f_J(t)dt - \left(\int_0^{\infty} t f_J(t)dt\right)^2$$

We use later the notation $q = 1 - \tilde{F}_R(t_\infty)$. Let us now compute the first term of the standard deviation:

$$\begin{aligned}
E(X^2) &= \int_0^{\infty} t^2 f_J(t)dt \\
&= \sum_{n=0}^{\infty} \int_{nt_\infty}^{(n+1)t_\infty} t^2 f_J^{[n,n+1]}(t)dt \\
&= \sum_{n=0}^{\infty} q^n \int_{nt_\infty}^{(n+1)t_\infty} t^2 \tilde{f}_R(t-nt_\infty)dt \quad \text{variable change: } u = t - nt_\infty \\
&= \sum_{n=0}^{\infty} q^n \int_0^{t_\infty} (u+nt_\infty)^2 \tilde{f}_R(u)du \\
&= \left(\sum_{n=0}^{\infty} q^n\right) \int_0^{t_\infty} u^2 \tilde{f}_R(u)du \\
&\quad + 2t_\infty \left(\sum_{n=0}^{\infty} nq^n\right) \int_0^{t_\infty} u\tilde{f}_R(u)du + t_\infty^2 \left(\sum_{n=0}^{\infty} n^2 q^n\right) \tilde{F}_R(t_\infty)
\end{aligned}$$

Using formulae from appendix A, we get:

$$\begin{aligned}
E(X^2) &= \frac{1}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty} u^2 \tilde{f}_R(u)du + \frac{2qt_\infty}{\tilde{F}_R(t_\infty)^2} \int_0^{t_\infty} u\tilde{f}_R(u)du + \frac{t_\infty^2 q(q+1)}{\tilde{F}_R(t_\infty)^3} \tilde{F}_R(t_\infty) \\
&= \frac{1}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty} u^2 \tilde{f}_R(u)du + 2\frac{t_\infty}{\tilde{F}_R(t_\infty)} \frac{(1-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty} u\tilde{f}_R(u)du \\
&\quad + \frac{t_\infty^2}{\tilde{F}_R(t_\infty)^2} \left(\tilde{F}_R(t_\infty)^2 - 3\tilde{F}_R(t_\infty) + 2\right)
\end{aligned}$$

Using equations 18 and 19, we get:

$$\int_0^{t_\infty} u^2 f(u)du = -t_\infty^2(1-\tilde{F}_R(t_\infty)) + 2\int_0^{t_\infty} u(1-\tilde{F}_R(u))du$$

and

$$\int_0^{t_\infty} uf(u)du = -t_\infty(1-\tilde{F}_R(t_\infty)) + \int_0^{t_\infty} (1-\tilde{F}_R(u))du$$

leading to:

$$\begin{aligned}
E(X^2) &= \frac{1}{\tilde{F_R}(t_\infty)}\left(-t_\infty^2(1-\tilde{F_R}(t_\infty)) + 2\int_0^{t_\infty} u(1-\tilde{F_R}(u))du\right)\\
&\quad +2\frac{t_\infty}{\tilde{F_R}(t_\infty)}\frac{(1-\tilde{F_R}(t_\infty))}{\tilde{F_R}(t_\infty)}\left(-t_\infty(1-\tilde{F_R}(t_\infty)) + \int_0^{t_\infty}(1-\tilde{F_R}(u))du\right)\\
&\quad +\frac{t_\infty^2}{\tilde{F_R}(t_\infty)^2}\left(\tilde{F_R}(t_\infty)^2 - 3\tilde{F_R}(t_\infty) + 2\right)\\
&= \frac{2}{\tilde{F_R}(t_\infty)}\int_0^{t_\infty} u(1-\tilde{F_R}(u))du) + 2\frac{t_\infty}{\tilde{F_R}(t_\infty)^2}(1-\tilde{F_R}(t_\infty))\int_0^{t_\infty}(1-\tilde{F_R}(u))du
\end{aligned}$$

And the second term:

$$E(X)^2 = \frac{1}{\tilde{F_R}(t_\infty)^2}\left(\int_0^{t_\infty}(1-\tilde{F_R}(u))du\right)^2$$

Combining $E(X^2)$ and $E(X)^2$, we get:

$$\begin{aligned}
\sigma_J(t_\infty) &= -\frac{1}{\tilde{F_R}(t_\infty)^2}\left(\int_0^{t_\infty}(1-\tilde{F_R}(u))du\right)^2 + \frac{2}{\tilde{F_R}(t_\infty)}\int_0^{t_\infty} u(1-\tilde{F_R}(u))du)\\
&\quad +2\frac{t_\infty}{\tilde{F_R}(t_\infty)^2}(1-\tilde{F_R}(t_\infty))\int_0^{t_\infty}(1-\tilde{F_R}(u))du
\end{aligned}$$

C Details on the computation of E_J in the case of the delayed resubmission strategy

In order not to have more than 2 identical jobs in the system at the same time, we impose $0 < t_0 < t_\infty$ and $(t_\infty - t_0) < t_0$ (this ensures that job 1 is canceled before job 3 is submitted). The probability for a single job to timeout is given by $q = 1 - \tilde{F_R}(t_\infty)$. If a job starts running at time t in the interval $[nt_0,\ (n-1)t_0 + t_\infty]$, this means that exactly $(n-1)$ jobs have timed out (probability q^{n-1}) and that either latency of job n is between t_0 and $(t-(n-1)t_0)$ (probability $(\tilde{F_R}(t-(n-1)t_0) - \tilde{F_R}(t_0)))$ or latency of job $n+1$ is lower than $(t - nt_0)$ (probability $\tilde{F_R}(t-nt_0)$). Since these last two events may both occur, the probability that at least one of them occur is equal to their sum minus their product:

$$
\begin{aligned}
F_J(t) &= P(J < t | t \in [nt_0,\ (n-1)t_0 + t_\infty]) \\
&= P(J < nt_0) + (1 - \tilde{F_R}(t_\infty))^{n-1} \big(\tilde{F_R}(t-(n-1)t_0) - \tilde{F_R}(t_0) + \tilde{F_R}(t - nt_0) \\
&\qquad - (\tilde{F_R}(t-(n-1)t_0) - \tilde{F_R}(t_0))\tilde{F_R}(t-nt_0)\big)
\end{aligned}
$$

Otherwise, if starts running at time t in the interval $[(n-1)t_0 + t_\infty,\ (n+1)t_0]$, this means that exactly n jobs have timed out (with probability q^n) and that the latency of job $n+1$ is lower than $t - nt_0$ (probability $\tilde{F_R}(t-nt_0)$).

$$
\begin{aligned}
F_J(t) &= P(J < t | t \in [(n-1)t_0 + t_\infty,\ (n+1)t_0]) \\
&= P(J < (n-1)t_0 + t_\infty) + (1 - \tilde{F_R}(t_\infty))^n \tilde{F_R}(t - nt_0)
\end{aligned}
$$

Deriving these last 2 equations leads to:

$$
\begin{cases}
\forall t \in [nt_0,\ (n-1)t_0 + t_\infty] & f_J(t) = q^{n-1}\Big(\tilde{f_R}(t-(n-1)t_0) \\
& \qquad + (1 + \tilde{F_R}(t_0))\tilde{f_R}(t - nt_0) \\
& \qquad - \tilde{f_R}(t-(n-1)t_0).\tilde{f_R}(t-nt_0)\Big) \\
\forall t \in [(n-1)t_0 + t_\infty,\ (n+1)t_0] & f_J(t) = q^n \tilde{f_R}(t - nt_0)
\end{cases}
$$

Finally, by integration, we get:

$$
\begin{aligned}
E_J &= \int_0^\infty t f_J(t)dt = \int_0^{t_0} t f_J(t)dt + \sum_{n=1}^{\infty}\left(\int_{nt_0}^{(n-1)t_0+t_\infty} t f_J(t)dt + \int_{(n-1)t_0+t_\infty}^{(n+1)t_0} t f_J(t)dt\right) \\
&= \int_0^{t_0} t\tilde{f}_R(t)dt + \sum_{n=1}^{\infty} q^{n-1}\int_{nt_0}^{(n-1)t_0+t_\infty} t\tilde{f}_R(t-(n-1)t_0)dt \\
&\quad +(1+\tilde{F}_R(t_0))\sum_{n=1}^{\infty} q^{n-1}\int_{nt_0}^{(n-1)t_0+t_\infty} t\tilde{f}_R(t-nt_0)dt \\
&\quad -\sum_{n=1}^{\infty} q^{n-1}\int_{nt_0}^{(n-1)t_0+t_\infty} t\tilde{f}_R(t-(n-1)t_0).\tilde{f}_R(t-nt_0)dt \\
&\quad +\sum_{n=1}^{\infty} q^{n}\int_{(n-1)t_0+t_\infty}^{(n+1)t_0} t\tilde{f}_R(t-nt_0)dt
\end{aligned}
$$

and replacing the integration variable t by $u = t - nt_0$ or $v = t - (n-1)t_0$, we obtain:

$$
\begin{aligned}
E_J &= \int_0^{t_0} t\tilde{f}_R(t)dt + \sum_{n=1}^{\infty} q^{n-1}\int_{t_0}^{t_\infty} (v+(n-1)t_0)\tilde{f}_R(v)dv \\
&+ (1+\tilde{F}_R(t_0))\sum_{n=1}^{\infty} q^{n-1}\int_0^{t_\infty-t_0} (u+nt_0)\tilde{f}_R(u)du + \sum_{n=1}^{\infty} q^{n}\int_{t_\infty-t_0}^{t_0} (u+nt_0)\tilde{f}_R(u)du \\
&- \sum_{n=1}^{\infty} q^{n-1}\int_0^{t_\infty-t_0} (u+nt_0)\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

Grouping terms and replacing the integer series by their values leads to:

$$
\begin{aligned}
E_J &= \int_0^{t_0} t\tilde{f}_R(t)dt \\
&+ \frac{1}{1-q}\int_{t_0}^{t_\infty} v\tilde{f}_R(v)dv + \frac{qt_0}{(1-q)^2}(\tilde{F}_R(t_\infty)-\tilde{F}_R(t_0)) \\
&+ (1+\tilde{F}_R(t_0))\left(\frac{1}{1-q}\int_0^{t_\infty-t_0} u\tilde{f}_R(u)du + \frac{t_0}{(1-q)^2}\tilde{F}_R(t_\infty-t_0)\right) \\
&+ \frac{q}{1-q}\int_{t_\infty-t_0}^{t_0} u\tilde{f}_R(u)du + \frac{qt_0}{(1-q)^2}(\tilde{F}_R(t_0)-\tilde{F}_R(t_\infty-t_0)) \\
&- \frac{t_0}{(1-q)^2}\int_0^{t_\infty-t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du - \frac{1}{1-q}\int_0^{t_\infty-t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

and replacing q by its value:

$$
\begin{aligned}
E_J &= \int_0^{t_0} t\tilde{f}_R(t)dt \\
&+ \frac{1}{\tilde{F}_R(t_\infty)}\int_{t_0}^{t_\infty} v\tilde{f}_R(v)dv + \frac{(1-\tilde{F}_R(t_\infty))t_0}{\tilde{F}_R(t_\infty)^2}(\tilde{F}_R(t_\infty)-\tilde{F}_R(t_0)) \\
&+ \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty-t_0} u\tilde{f}_R(u)du + \frac{\tilde{F}_R(t_0)}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty-t_0} u\tilde{f}_R(u)du \\
&+ (1+\tilde{F}_R(t_0))\tfrac{t_0}{\tilde{F}_R(t_\infty)^2}\tilde{F}_R(t_\infty-t_0) \\
&+ \frac{1}{\tilde{F}_R(t_\infty)}\int_{t_\infty-t_0}^{t_0} u\tilde{f}_R(u)du - \int_{t_\infty-t_0}^{t_0} u\tilde{f}_R(u)du + \frac{(1-\tilde{F}_R(t_\infty))t_0}{\tilde{F}_R(t_\infty)^2}(\tilde{F}_R(t_0)-\tilde{F}_R(t_\infty-t_0)) \\
&- \frac{t_0}{\tilde{F}_R(t_\infty)^2}\int_0^{t_\infty-t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du - \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty-t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

And finally:

$$
\boxed{
\begin{aligned}
E_J(t_0,t_\infty) &= \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty} u\tilde{f}_R(u)du + \frac{\tilde{F}_R(t_0)}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty-t_0} u\tilde{f}_R(u)du \\
&+\frac{t_0}{\tilde{F}_R(t_\infty)} + t_0\frac{\tilde{F}_R(t_\infty-t_0)}{\tilde{F}_R(t_\infty)} + t_0\frac{\tilde{F}_R(t_0)\tilde{F}_R(t_\infty-t_0)}{\tilde{F}_R^{\,2}(t_\infty)} - t_0 + \int_0^{t_\infty-t_0} u\tilde{f}_R(u)du \\
&-\frac{t_0}{\tilde{F}_R(t_\infty)^2}\int_0^{t_\infty-t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du - \frac{1}{\tilde{F}_R(t_\infty)}\int_0^{t_\infty-t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
}
\tag{20}
$$

D Computation of σ in the case of the delayed re-submission strategy

Back to the definition:

$$\text{var}(X) = E(X^2) - (E(X))^2 = \int_0^\infty t^2 f_J(t)dt - \left(\int_0^\infty t f_J(t)dt\right)^2$$

We first compute $E(X^2)$:

$$\begin{aligned}
E(X^2) &= \int_0^\infty t^2 f_J(t)dt \\
&= \int_0^{t_0} t^2 f_J(t)dt + \sum_{n=1}^\infty \left(\int_{nt_0}^{(n-1)t_0+t_\infty} t^2 f_J(t)dt + \int_{(n-1)t_0+t_\infty}^{(n+1)t_0} t^2 f_J(t)dt\right) \\
&= \int_0^{t_0} t^2 \tilde{f}_R(t)dt + \sum_{n=1}^\infty q^{n-1} \int_{nt_0}^{(n-1)t_0+t_\infty} t^2 \tilde{f}_R(t-(n-1)t_0)dt \\
& +(1+\tilde{F}_R(t_0)) \sum_{n=1}^\infty q^{n-1} \int_{nt_0}^{(n-1)t_0+t_\infty} t^2 \tilde{f}_R(t-nt_0)dt \\
& -\sum_{n=1}^\infty q^{n-1} \int_{nt_0}^{(n-1)t_0+t_\infty} t^2 \tilde{f}_R(t-(n-1)t_0).\tilde{f}_R(t-nt_0)dt \\
& +\sum_{n=1}^\infty q^n \int_{(n-1)t_0+t_\infty}^{(n+1)t_0} t^2 \tilde{f}_R(t-nt_0)dt
\end{aligned}$$

and replacing the integration variable t by $u = t - nt_0$ or $v = t - (n-1)t_0$, we obtain:

$$\begin{aligned}
E(X^2) &= \int_0^{t_0} t^2 \tilde{f}_R(t)dt + \sum_{n=1}^\infty q^{n-1} \int_{t_0}^{t_\infty} (v+(n-1)t_0)^2 \tilde{f}_R(v)dv \\
&+ (1+\tilde{F}_R(t_0)) \sum_{n=1}^\infty q^{n-1} \int_0^{t_\infty - t_0} (u+nt_0)^2 \tilde{f}_R(u)du \\
&+ \sum_{n=1}^\infty q^n \int_{t_\infty - t_0}^{t_0} (u+nt_0)^2 \tilde{f}_R(u)du \\
&- \sum_{n=1}^\infty q^{n-1} \int_0^{t_\infty - t_0} (u+nt_0)^2 \tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}$$

isolating the terms in the sums leads to:

$$
\begin{aligned}
E(X^2) &= \int_0^{t_0} t^2 \tilde{f}_R(t)dt + \sum_{n=1}^{\infty} q^{n-1} \int_{t_0}^{t_\infty} v^2 \tilde{f}_R(v)dv + t_0^2 \sum_{n=1}^{\infty} q^{n-1}(n-1)^2 \int_{t_0}^{t_\infty} \tilde{f}_R(v)dv \\
&+ 2t_0 \sum_{n=1}^{\infty} (n-1)q^{n-1} \int_{t_0}^{t_\infty} v\tilde{f}_R(v)dv \\
&+ (1+\tilde{F}_R(t_0)) \sum_{n=1}^{\infty} q^{n-1} \int_0^{t_\infty - t_0} u^2 \tilde{f}_R(u)du \\
&+ (1+\tilde{F}_R(t_0))t_0^2 \sum_{n=1}^{\infty} n^2 q^{n-1} \int_0^{t_\infty - t_0} \tilde{f}_R(u)du \\
&+ 2t_0(1+\tilde{F}_R(t_0)) \sum_{n=1}^{\infty} nq^{n-1} \int_0^{t_\infty - t_0} u\tilde{f}_R(u)du \\
&+ \sum_{n=1}^{\infty} q^n \int_{t_\infty - t_0}^{t_0} u^2 \tilde{f}_R(u)du + t_0^2 \sum_{n=1}^{\infty} n^2 q^n \int_{t_\infty - t_0}^{t_0} \tilde{f}_R(u)du \\
&+ 2t_0 \sum_{n=1}^{\infty} nq^n \int_{t_\infty - t_0}^{t_0} u\tilde{f}_R(u)du - \sum_{n=1}^{\infty} q^{n-1} \int_0^{t_\infty - t_0} u^2 \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- t_0^2 \sum_{n=1}^{\infty} n^2 q^{n-1} \int_0^{t_\infty - t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- 2t_0 \sum_{n=1}^{\infty} nq^{n-1} \int_0^{t_\infty - t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

and replacing the series:

$$
\begin{aligned}
E(X^2) &= \int_0^{t_0} t^2 \tilde{f}_R(t)dt + \frac{1}{\tilde{F}_R(t_\infty)} \int_{t_0}^{t_\infty} v^2 \tilde{f}_R(v)dv \\
&+ t_0^2 \tfrac{(2-\tilde{F}_R(t_\infty))(1-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^3} \int_{t_0}^{t_\infty} \tilde{f}_R(v)dv \\
&+ 2t_0 \frac{(1-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^2} \int_{t_0}^{t_\infty} v\tilde{f}_R(v)dv \\
&+ (1+\tilde{F}_R(t_0)) \frac{1}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty - t_0} u^2 \tilde{f}_R(u)du \\
&+ (1+\tilde{F}_R(t_0)) t_0^2 \tfrac{(2-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^3} \int_0^{t_\infty - t_0} \tilde{f}_R(u)du \\
&+ 2t_0(1+\tilde{F}_R(t_0)) \frac{1}{\tilde{F}_R(t_\infty)^2} \int_0^{t_\infty - t_0} u\tilde{f}_R(u)du \\
&+ \frac{1-\tilde{F}_R(t_\infty)}{\tilde{F}_R(t_\infty)} \int_{t_\infty - t_0}^{t_0} u^2 \tilde{f}_R(u)du + t_0^2 \frac{(1-\tilde{F}_R(t_\infty))(2-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^3} \int_{t_\infty - t_0}^{t_0} \tilde{f}_R(u)du \\
&+ 2t_0 \frac{(1-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^2} \int_{t_\infty - t_0}^{t_0} u\tilde{f}_R(u)du \\
&- \frac{1}{\tilde{F}_R(t_\infty)} \int_0^{t_\infty - t_0} u^2 \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- t_0^2 \frac{(1-\tilde{F}_R(t_\infty))(2-\tilde{F}_R(t_\infty))}{\tilde{F}_R(t_\infty)^3} \int_0^{t_\infty - t_0} \tilde{f}_R(u+t_0).\tilde{f}_R(u)du \\
&- 2t_0 \frac{1}{\tilde{F}_R(t_\infty)^2} \int_0^{t_\infty - t_0} u\tilde{f}_R(u+t_0).\tilde{f}_R(u)du
\end{aligned}
$$

E Detail on the computation of $N_{//}$ in the case of the delayed resubmission strategy

With the delayed resubmission strategy, a variable number of jobs may be running on the infrastructure at any time. Let $N_{//}$ denote the average number of jobs needed. $N_{//}$ is to be compared with the value of b in the case of multiple resubmissions. Let l be the latency of a job with parameters t_0 and t_∞. We can notice that, after the first period $[0;t_0]$, the number of jobs in parallel is periodic with a t_0-period of time. In the following, let n be an integer such that l is in $[nt_0,(n+1)t_0[$.

E.1 Computation of $N_{//}$

We gives here the details for the case where $\mathbf{n > 1}$.

For both cases detailed below, only one job is running until t_0. Then, for $(n-1)$ period of t_0 time, we will have 2 jobs in parallel during $|I_0| = (t_\infty - t_0)$ and only 1 job during $|I_1| = (t_0 - (t_\infty - t_0)) = 2t_0 - t_\infty$ which leads to $2*(t_\infty - t_0) + 2t_0 - t_\infty = t_\infty$.

- if $l \in [nt_0; (n-1)t_0 + t_\infty[$ (I_0 in figure 5): after the $(n-1)t_0$ first periods (where $2(t_\infty - t_0) + 2t_0 - t_\infty)/t_0 = t_\infty/t_0$ jobs are running in parallel), 2 jobs are running in parallel during $(l - nt_0)$ which leads to :

$$N_{//} = \frac{t_0 + (n-1)t_\infty + 2(l - nt_0)}{l} = 2 + \frac{(n-1)t_\infty - (2n-1)t_0}{l}$$

- if $l \in [(n-1)t_0 + t_\infty; (n+1)t_0[$ (I_1 in figure 5): after the $(n-1)t_0$ first periods, 2 jobs are running in parallel during $(t_\infty - t_0)$ and then only one job during $(l - ((n-1)t_0 + t_\infty))$ which leads to :

$$N_{//} = \frac{t_0 + (n-1)t_\infty + 2(t_\infty - t_0) + (l - ((n-1)t_0 + t_\infty))}{l} = 1 + n\frac{t_\infty - t_0}{l}$$

These two expressions are still valid in the case of $n = 1$.

E.2 Variations of $N_{//}$

if $l \in [nt_0; (n-1)t_0 + t_\infty[$

$$\frac{\partial N_{//}}{\partial l} = -\frac{(n-1)t_\infty - (2n-1)t_0}{l^2} > 0$$

Thus $N_{//}$ is increasing with respect to l.

$$N_{//min} = N_{//}(nt_0) = \frac{1}{n} + \frac{t_\infty}{t_0}(1 - \frac{1}{n}) \in [1\ ;\ 2 - \frac{1}{n}]$$

$$N_{//max} = N_{//}((n-1)t_0 + t_\infty) = 2 + \frac{(n-1)t_\infty - (2n-1)t_0}{(n-1)t_0 + t_\infty} \in [1\ ;\ 2 - \frac{1}{n+1}]$$

if $l \in [(n-1)t_0 + t_\infty; (n+1)t_0[$

$$\frac{\partial N_{//}}{\partial l} = -n\frac{t_\infty - t_0}{l^2} < 0$$

$N_{//}$ is thus decreasing with respect to l.

$$N_{//min} = N_{//}((n+1)t_0) = \frac{t_\infty}{t_0} - (\frac{t_\infty}{t_0} - 1)\frac{1}{n+1} \in [1\ ;\ 2 - \frac{1}{n+1}]$$

$$N_{//max} = N_{//}((n-1)t_0 + t_\infty) = 1 + n\frac{t_\infty - t_0}{(n-1)t_0 + t_\infty} \in [1\ ;\ 2 - \frac{1}{n+1}]$$

Conclusion From this study, we easily conclude that:

$$N_{//max} \in [1\ ;\ 2 - \frac{1}{n+1}]$$

E.3 Functions used for the study of the variations of $N_{//}$

We detail in this paragraph the determination of intervals for the minimal and maximal values of $N_{//}$. Taking into account the fact that $\frac{t_\infty}{t_0} \in [1\ ;\ 2]$, we pose $x = \frac{t_\infty}{t_0}$ and study the variations of the following functions.

function f

$$f(x) = \frac{1}{n} + x(1 - \frac{1}{n})$$

$$f'(x) = 1 - \frac{1}{n} > 0$$

f is thus increasing and:

$$f_{min} = f(1) = 1$$

$$f_{max} = f(2) = 2 - \frac{1}{n}$$

function g

$$g(x) = 2 + \frac{(n-1)x - 2n + 1}{x + n - 1}$$

$$g'(x) = \frac{n^2}{(x+n-1)^2} > 0$$

g is thus increasing and:

$$g_{min} = g(1) = 1$$

$$g_{max} = g(2) = 2 - \frac{1}{n+1}$$

function h

$$h(x) = x - (x-1)\frac{1}{n+1}$$

$$h'(x) = \frac{n}{n+1} > 0$$

h is thus increasing and:

$$h_{min} = h(1) = 1$$

$$h_{max} = h(2) = 2 - \frac{1}{n+1}$$

function k

$$k(x) = 1 + n\frac{x-1}{x+n-1}$$

$$k'(x) = \frac{n^2}{(x+n-1)^2} > 0$$

k is thus increasing and:

$$k_{min} = k(1) = 1$$

$$k_{max} = k(2) = 2 - \frac{1}{n+1}$$